

Esame di Calcolatori Elettronici T

4 Febbraio 2020 (Ing. Informatica)

Esercizio 1

Progettare un sistema basato sul **processore DLX**, dotato di **1 GB di EPROM agli indirizzi bassi e 1 GB di RAM agli indirizzi alti**.

Nel sistema sono presenti una porta in input e una porta in output, già progettate e denominate rispettivamente **INPUT_PORT** e **OUTPUT_PORT**, che comunicano con l'esterno mediante il protocollo di *handshake*. **I dati letti da INPUT_PORT dovranno essere scritti all'indirizzo FFFF0020h mentre a OUTPUT_PORT dovrà essere inviato, quando questa operazione è possibile, il dato letto da INPUT_PORT. L'invio del dato a OUTPUT_PORT, deve avvenire (quando possibile) contemporaneamente alla lettura da INPUT_PORT.** Non è prevista altra modalità per eseguire trasferimenti verso OUTPUT_PORT. Infine, **è necessario contare modulo 2^{16} , mediante una opportuna rete logica, il numero di trasferimenti contemporanei sulle due porte e poter leggere tale valore via software.** All'avvio il conteggio deve essere inizializzato al valore 0.

- Per prima cosa, **descrivere sinteticamente la soluzione** che s'intende realizzare e indicare **chiaramente quali sono i dispositivi utilizzati e segnali di chip-select**
- Progettare il sistema in base alle specifiche del testo, **minimizzando le risorse necessarie ed evidenziando e gestendo eventuali criticità**
- Indicare le espressioni di decodifica e il range di indirizzi di tutte le periferiche, le memorie e i segnali
- Scrivere il codice dell'interrupt handler che gestisce i trasferimenti con INPUT_PORT assumendo che i registri da R20 a R25 possano essere utilizzati senza la necessità di doverli ripristinare
- Scrivere il codice per leggere, nel registro R7, il valore di conteggio dei trasferimenti contemporanei effettuati

Esercizio 2 \rightarrow Permette di anticipare il registro in IO/EX ed evitare stalli (FORWARDING UNIT)
A cosa serve il multiplexer posto sull'uscita del *register file* nel DLX pipelined?
Ci sono due MUX connessi ai registri A e B, uno per le istruzioni I (registro + immediato) e uno per i salti (PC + 4 + indirizzo)

Esercizio 3

- a) Quali strategie è opportuno adottare nella gestione dei registri nella scrittura di un *interrupt handler*? *Vengono salvati e ripristinati prima di RFE*
- b) Cosa può accadere se non si adottano tali strategie?

Risposte vaghe e/o non focalizzate sulle domande del testo non saranno MINIMAMENTE considerate.

Come noto è cambiata la procedura di verbalizzazione e il periodo di validità dei voti. Si consulti il sito del corso per maggiori dettagli.

CNIP SELECT

1 GB EPROM		1 GB RAM
------------	--	----------

CS_EPROM_0 = BA31 BA30 B60
 " -1 = B61
 " -2 = B62
 " -3 = B63

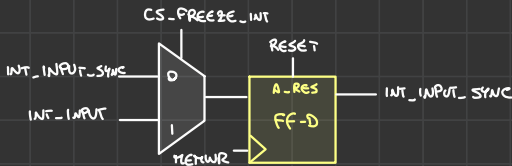
CS_RAM_0 = BA31 BA29 B60
 " -1 = B61
 " -2 = B62
 " -3 = B63

CS_RAM_N_0 = BA31 BA29 B60
 " -1 = B61
 " -2 = B62
 " -3 = B63

CS_INPUT_PORT = BA31 BA30 B60
 CS_FREEZE_INT = B61

CS_READ_COUNTER = B62, B63 · MEMRD

SINCRONIZZAZIONE DEL SEGNALE INT INPUT



Perché le porte in output deve essere sincronizzate con quelle di input

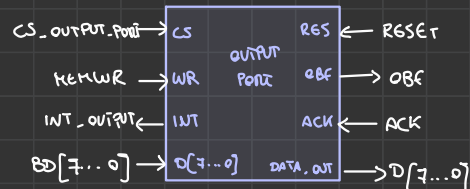
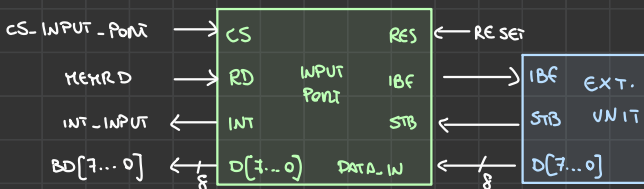
$$CS_OUTPUT_PORT = INT_INPUT_SYNC \cdot CS_INPUT_PORT$$

$$INT(TO CLK) = INT_INPUT$$

CONTATORE



PORTE DI I/O



CODICE

INTERRUPT HANDLER

```
0h handler: LHI R20, 0x4000 ; R20 = 0x40000000
4h          SB R0, 0x0001(R20); Scritture dummy a CS-FREEZE_INT
8h          LB R21, 0x0000(R20); R21 =  $\pi[R20]$ 
Ch          LHI R22, 0xFFFF ; R22 = 0xFFFF0000
10h         SW R21, 0x0020(R22); Scrivo R21 a 0xFFFF0020
14h         RFG ; Torneo da interrupt
```

LEGGERE DA COUNTER

```
100h LHI R8, 0x4000 ; R8 = 0x40000000
104h LHU R7, 0x0002(R8);  $R7 \leftarrow \frac{1}{16} \pi[R8 + 0x0002]$ 
```